# 1 MDPs: Basic Conceptual Questions

(a) In class, we learned that the Bellman Equations can be used to characterize optimal utility in MDPs. For reference, recall that this equation is given as:

$$V^*(s) = \max_a \sum_{s'} T(s, a, s')[R(s, a, s') + \gamma V^*(s')]$$

What do we call $\gamma$ in this equation? Why is it necessary? What happens as $\gamma$ grows larger? As it grows smaller?

(b) What are the key distinctions between the value iteration and policy iteration algorithms, and when might you prefer one to the other?

(c) When does policy iteration end? Immediately after policy iteration ends (without performing additional computation), do we have the values of the optimal policy?

(d) What changes if during policy iteration, you only run one iteration of Bellman update instead of running it until convergence? Do you still get an optimal policy?

# 2 MDPs: Racing

Consider a modification of the racing robot car example seen in lecture. In this game, the car repeatedly moves a random number of spaces that is equally likely to be 2, 3, or 4. The car can either Move or Stop if the total number of spaces moved is less than 6.

If the total spaces moved is 6 or higher, the game automatically ends, and the car receives a reward of 0. When the car Stops, the reward is equal to the total spaces moved (up to 5), and the game ends. There is no reward for the Move action.

Let's formulate this problem as an MDP with the states $\{0, 2, 3, 4, 5, Done\}$.

(a) What is the transition function for this MDP? (You should specify discrete values for specific state/action inputs.)

(b) What is the reward function for this MDP?

(c) Recall the value iteration update equation:

$$V_{k+1}(s) = \max_a \sum_{s'} T(s, a, s')[R(s, a, s') + \gamma V_k(s')]$$

Perform value iteration for 4 iterations with $\gamma = 1$.

| States | 0 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $V_0$ | | | | | |
| $V_1$ | | | | | |
| $V_2$ | | | | | |
| $V_3$ | | | | | |
| $V_4$ | | | | | |

(d) You should have noticed that value iteration converged above. What is the optimal policy?

| States | 0 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\pi^*$ | | | | | |

(e) How would our results change with $\gamma = 0.1$?

(f) Now recall the policy evaluation and policy improvement equations, which together make up policy iteration. Bellman Equation for policy evaluation:

$$V^\pi_{k+1}(s) = \sum_{s'} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V^\pi_k(s')]$$

Policy improvement:

$$\pi_{new}(s) \leftarrow \underset{a}{\operatorname{argmax}} \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^{\pi_{old}}(s')]$$

Perform two iterations of policy iteration for one step of this MDP, starting from the fixed policy below. Use the initial $\gamma = 1$.

| States | 0 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\pi_0$ | Move | Stop | Move | Stop | Move |
| $V^{\pi_0}$ | | | | | |
| $\pi_1$ | | | | | |
| $V^{\pi_1}$ | | | | | |
| $\pi_2$ | | | | | |